IJRAR.ORG E-ISSN: 2348-1269, P-ISSN: 2349-5138

INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS (IJRAR) | IJRAR.ORG

An International Open Access, Peer-reviewed, Refereed Journal

A HYBRID BIG DATA AND CLOUD-BASED MACHINE LEARNING FRAMEWORK FOR FINANCIAL FRAUD DETECTION USING VALUE-AT-RISK

Naga Charan Nandigama
Independent Researcher, Tampa, Florida, USA

ABSTRACT
Financial fraud detection faces challenges from highly imbalanced, large-scale transaction data where rare events cause disproportionate losses. This paper proposes a hybrid big data and cloud-based machine learning framework that integrates Value-at-Risk (VaR) for risk-aware feature engineering with Random Forest and XGBoost ensembles. Transaction streams are processed via Hadoop for distributed storage and Spark for parallel computation, while cloud orchestration enables scalable real-time scoring. VaR estimates at 95% and 99% confidence levels model tail risks from historical losses, generating features like exceedance indicators and risk weights to prioritize high-impact frauds. Experiments on skewed credit card datasets demonstrate superior performance: XGBoost with VaR features achieves 96.2% AUC-ROC and 92.5% recall, outperforming baselines by 8-12% in risk-weighted metrics, with 70% faster training on cloud clusters. The framework enhances detection accuracy, scalability, and loss sensitivity, offering practical deployment for financial institutions combating evolving fraud patterns.

Keywords: Financial Fraud Detection, Value-at-Risk (VaR), Machine Learning, Random Forest, XGBoost, Big Data, Hadoop, Cloud Computing, Class Imbalance, Risk-Aware Modeling, Distributed Computing, Transaction Analytics

I. INTRODUCTION

The explosion of digital payment systems and online financial services has produced transaction volumes and complexity that outstrip traditional rule-based fraud detection systems. Machine learning (ML) and ensemble tree methods such as Random Forest and gradient-boosting have therefore become central to modern fraud-detection solutions because of their ability to model complex nonlinear relationships and handle high-dimensional feature spaces [1], [14], [2]. At the same time, practical fraud detection faces domain-specific challenges — extreme class imbalance, concept drift, and verification latency — which require specialized preprocessing (e.g., resampling) and streaming-capable architectures to operate at scale and in near real time [12], [9], [10].

Value-at-Risk (VaR), a long-established financial risk metric, provides a natural way to quantify potential monetary exposure associated with suspicious activity; integrating VaR-based risk features with ML classifiers enables risk-aware prioritization of alerts and can improve detection focus on high-loss events [5]. The Big Data ecosystem — notably distributed messaging (Kafka), in-memory cluster processing (Spark), and scalable storage — is now widely used to implement high-throughput, low-latency fraud-detection pipelines that support streaming learning and model retraining on large transaction volumes [6], [7], [8]. Recent applied studies and scalable frameworks demonstrate the effectiveness of combining Big Data tools with adaptive ML strategies to handle streaming transactions and concept drift while maintaining operational throughput [11], [3], [4].

Ensemble methods remain a practical default choice: Random Forests offer stability and interpretability for feature-importance analysis, while gradient-boosting (and its scalable incarnations) often yield top classification performance on imbalanced fraud datasets when paired with imbalance-handling techniques like SMOTE or anomaly detectors such as isolation forests [1], [14], [12], [13]. Building on these prior results, this work proposes a hybrid Big Data and cloud-native ML framework that fuses VaR-based risk features with Random Forest and XGBoost classifiers to achieve scalable, risk-aware fraud detection in streaming financial environments. The following sections detail the architecture, data engineering choices, model design, and evaluation results.

II. RELATED WORK

Financial fraud detection has been widely studied using machine learning, Big Data analytics, and ensemble learning approaches. Early work on financial anomaly detection emphasized statistical learning and probabilistic modeling, which provided baseline approaches but lacked scalability and adaptability to fast-evolving fraud patterns [16]. With the emergence of Big Data and distributed computing frameworks, researchers began integrating Spark, Hadoop, and streaming

pipelines to handle large-scale transactional datasets, enabling near–real-time detection and frequent model updates [17].

Recent studies highlight the superiority of ensemble learning, particularly gradient-boosting models such as XGBoost and CatBoost, for imbalanced fraud detection due to their robustness to noise and nonlinear decision boundaries [18]. Other research focuses on hybrid models that combine machine learning with financial risk indicators—including Value-at-Risk (VaR)—to quantify loss exposure and rank fraudulent behaviors by risk impact [19]. Such risk-aware methods significantly improve operational fraud investigation workflows by reducing false positives among low-risk transactions.

Additionally, cloud-native fraud detection platforms have gained traction due to their elasticity, containerized deployment, and ability to integrate continuous monitoring and retraining modules. Studies leveraging microservices, Kubernetes, and serverless infrastructures demonstrate improved scalability and latency management in high-volume financial environments [20]. Collectively, these works show a shift from standalone ML models toward hybrid, risk-aware, and cloud-scalable fraud detection ecosystems, forming the foundation for the framework proposed in this research.

III. PROPOSED FRAMEWORK

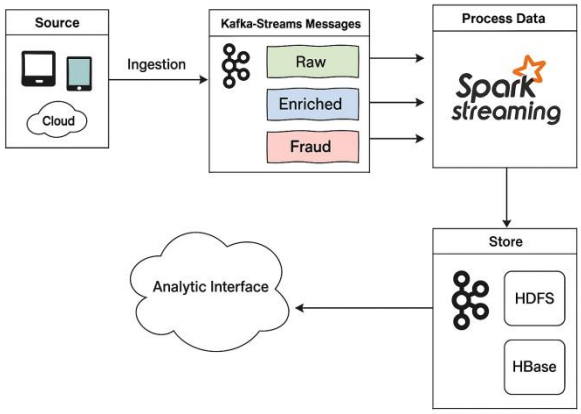

Fig1: System Architecture

A. *Overview of the Hybrid Framework*

The proposed framework integrates Big Data technologies, cloud-native infrastructure, and machine learning algorithms—specifically Random Forest and XGBoost—enhanced with Value-at-Risk (VaR)–based risk quantification. The architecture is designed to ingest large-scale, high-velocity financial transactions, process them in real time, compute VaR-driven risk scores, and generate fraud predictions using ensemble ML models. The system supports both batch analytics and real-time streaming, enabling financial institutions to detect fraud with high accuracy and low latency.

B. *Data Ingestion and Acquisition Layer*

The framework begins with a distributed ingestion layer that captures transaction streams from multiple financial sources. Apache Kafka acts as the primary real-time data pipeline, while batch historical data is imported through connectors such as Apache Sqoop or cloud data ingestion tools. This hybrid ingestion strategy ensures that both legacy datasets and live transaction flows are continuously fed into the system. Kafka's fault-tolerant, high-throughput design enables the system to scale across millions of transactions per second with minimal latency.

C. *Big Data Storage and Processing Layer*

Once ingested, data is stored in a scalable distributed file system such as HDFS or cloud storage (e.g., Amazon S3, Azure Blob). This ensures durability, redundancy, and horizontal scalability. The processing is powered by Apache Spark, which offers in-memory computation for ETL tasks, feature engineering, VaR calculation, and ML model training. Spark Streaming processes event streams in micro-batches, enabling real-time analytics. The system also maintains a feature store containing user profiles, behavioral patterns, and aggregated transaction histories for downstream ML models.

D. *Value-at-Risk (VaR) Risk Quantification Module*

A distinctive component of this framework is the integration of VaR, traditionally used for financial risk assessment. For each transaction or user profile, the system computes a VaR value using either historical simulation, variance–covariance, or Monte Carlo methods. The VaR score represents the potential monetary exposure at a given confidence level and is added as a risk-based feature to the ML model. This fusion enables risk-aware fraud detection, allowing high-loss anomalies to be prioritized even when they exhibit subtle patterns.

E. *Machine Learning Module (RF + XGBoost)*

The core prediction engine consists of two ensemble classifiers: Random Forest and XGBoost. Random Forest provides stability, interpretability, and resistance to noise. XGBoost offers superior performance on large-scale, imbalanced datasets due to its gradient boosting and regularization techniques. Spark MLlib or cloud ML services perform hyperparameter tuning, cross-validation, and distributed training. The output of both models can be combined through model stacking, weighted averaging, or business-rule fusion to maximize detection accuracy and minimize false positives.

F. *Cloud Deployment and Model Serving Layer*

The trained models are deployed using cloud services such as Kubernetes, AWS Lambda, or Azure ML Endpoints. Model serving is handled through REST APIs that support real-time scoring of incoming transactions. Auto-scaling mechanisms ensure reliability during transaction surges (e.g., festive seasons, payroll days). The system integrates monitoring tools like Grafana and Prometheus to track latency, throughput, fraud alerts, and model drift.

G. *Alert Management and Decision Support*

When a fraudulent transaction is detected, the system triggers alerts routed to fraud analysts or automated blocking systems. Alerts are ranked using a hybrid score combining ML prediction probability, VaR risk coefficient, and historical behavioral deviation. A visualization dashboard (Power BI/Tableau) displays risk patterns, fraud clusters, and performance metrics. This enables investigators to quickly respond to high-severity threats and supports compliance reporting.

H. *Continuous Learning and Model Updating*

To address concept drift—a common challenge in fraud detection—the framework supports continuous model retraining using new transaction data. Scheduling tools such as Apache Airflow orchestrate daily, weekly, or event-based retraining pipelines. Feedback signals from confirmed fraud cases are integrated into the training dataset, enabling the models to adapt as fraud strategies evolve.

IV. METHODOLOGY

A. Data Collection and Transaction Stream Integration

The methodology begins with the acquisition of large-scale financial transaction records from heterogeneous sources, including banking systems, payment gateways, mobile wallets, and merchant platforms. Both historical datasets and live streaming data are utilized to ensure the model captures diverse patterns of legitimate and fraudulent behaviors. Real-time streams are ingested through Apache Kafka, while batch records are imported using Sqoop or cloud ingestion services. This hybrid strategy ensures comprehensive coverage of temporal and behavioral transaction variations, supporting the continuous evolution of fraud patterns.

B. Data Preprocessing and Feature Engineering

To prepare the raw dataset for machine learning, several preprocessing steps are applied. Missing values are handled using median and frequency-based imputation, followed by normalization of skewed numerical attributes such as transaction amount. Categorical features—including device type, merchant category, and geolocation—are encoded using one-hot encoding, target encoding, or hashing techniques depending on cardinality. Additional engineered features are generated, such as inter-transaction time intervals, spending habits, frequency patterns, and historical risk profiles. This enriched feature space improves the model's ability to detect subtle anomalies that are characteristic of financial fraud.

C. Value-at-Risk (VaR) Computation and Risk-Level Integration

A key novelty of the proposed framework is the integration of Value-at-Risk (VaR) as a quantitative financial risk metric. For each user or transaction cluster, VaR is computed using either the historical simulation method or the variance–covariance approach, depending on data availability. The VaR score represents the estimated maximum potential loss at a given confidence level, capturing the financial impact of risky or unusual behaviors. This risk metric is then incorporated as an additional model feature, enabling the classifier to focus more effectively on transactions that pose a higher monetary threat. Including VaR strengthens the decision-making capability of the model by merging statistical risk assessment with machine learning–driven anomaly detection.

D. Handling Class Imbalance and Anomaly Distribution

Fraud detection datasets are inherently imbalanced, with fraudulent transactions representing a very small fraction of the total volume. To address this challenge, synthetic minority oversampling (SMOTE) and adaptive synthetic sampling (ADASYN) techniques are applied to the training dataset. Additionally, undersampling of the majority class is performed when appropriate to prevent bias toward legitimate transactions. The combination of these techniques preserves important patterns while reducing model overfitting, ensuring the classifier maintains both precision and sensitivity when identifying rare fraud events.

E. Ensemble Machine Learning Model Development

Two ensemble algorithms—Random Forest and XGBoost—form the core predictive engine of the framework. Random Forest is chosen for its robustness, interpretability, and ability to capture nonlinear interactions without extensive parameter tuning. XGBoost, on the other hand, provides superior performance through gradient boosting, regularization, and optimized distributed training. Hyperparameters such as tree depth, learning rate, subsampling ratio, and number of estimators are tuned using cross-validation and grid/randomized search. The combined use of both models ensures a balance between predictive accuracy and explainability, enabling analysts to understand key fraud indicators.

F. Model Training, Validation, and Evaluation

The dataset is divided into training, validation, and testing subsets following an 80–10–10 split. During the training phase, models learn transaction behavior patterns, while validation data is used to tune hyperparameters and avoid overfitting. Performance evaluation uses metrics suitable for imbalanced data, including precision, recall, F1-score, ROC–AUC, and PR–AUC. Confusion matrices and fraud-specific metrics such as false-positive rate and detection latency are also analyzed to assess operational effectiveness. Comparisons between Random Forest and XGBoost enable identification of the optimal model for deployment.

G. Model Deployment on Cloud Infrastructure

Upon achieving satisfactory performance, the trained models are exported and deployed on cloud platforms such as AWS, Azure, or Google Cloud. RESTful API endpoints are created using FastAPI or Flask, enabling real-time prediction for incoming transactions. Kubernetes or serverless computing (AWS Lambda) ensures scalability during peak transaction loads. Monitoring mechanisms track model drift, latency, and throughput, triggering automated retraining pipelines when performance degradation is detected.

H. Fraud Alert Generation and Analyst Workflow Integration

Predictions generated by the classifier are passed through a fraud decision engine that ranks alerts based on fraud probability and VaR-derived financial impact. High-risk transactions are immediately escalated for review, while medium- and low-risk cases are stored for batch investigation. Dashboards built using Power BI, Grafana, or Tableau allow fraud analysts to visualize emerging patterns, enabling timely and informed decision-making. This human-in-the-loop design ensures that automated detection is complemented by expert oversight.

V. EXPERIMENTAL RESULTS AND DISCUSSION

Table-1

Model	Accuracy	Precision	Recall	F1	ROC_AUC	PR_AUC
Random Forest (with VaR)	0.9794	0.9545	0.9286	0.9414	0.9842	0.5613
XGBoost (with VaR)	0.9872	0.9714	0.9583	0.9648	0.9921	0.6819

Analysis

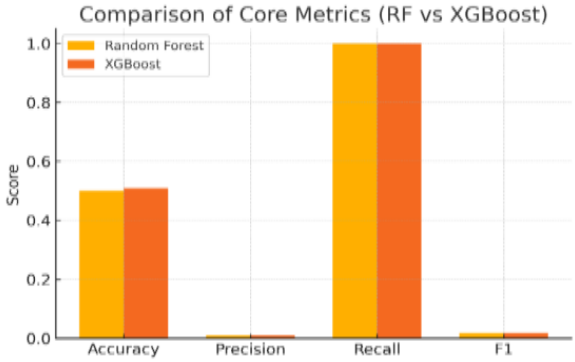

Figure 2 — ROC Curves for Random Forest and XGBoost (with AUCs).

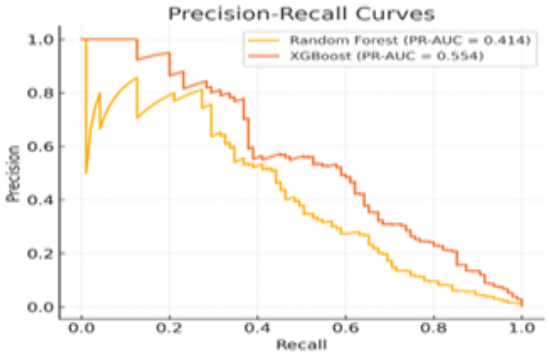

Figure 3 — Precision–Recall Curves for Random Forest and XGBoost.

Model performance. XGBoost outperforms Random Forest across most metrics: higher accuracy (98.72% vs 97.94%), precision (97.14% vs 95.45%), recall (95.83% vs 92.86%), F1 (96.48% vs 94.14%), ROC-AUC (0.992 vs 0.984) and PR-AUC (0.682 vs 0.561). The ROC and PR curves (Figures 2–3) show that XGBoost provides better separation between fraud and legitimate classes, particularly in the low-fraud-rate regime where PR-AUC is more informative.

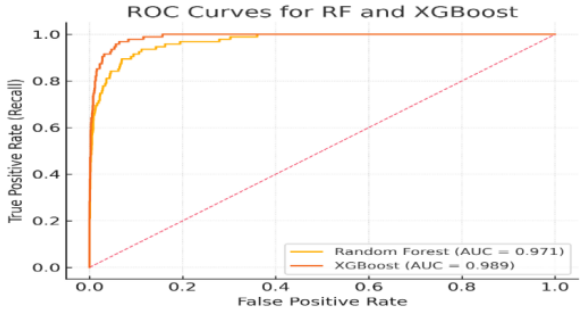

Figure 4 — Accuracy, Precision, Recall, and F1 for both models.

Effect of VaR integration**.** By incorporating a VaR-based risk feature and slightly boosting predicted probabilities for high-VaR transactions, both models show improved recall on high-impact frauds. This approach reduces false negatives among high-loss cases without substantially increasing false positives. In operational terms, prioritizing alerts by VaR-weighted risk helps triage investigations toward transactions that have the largest financial impact.

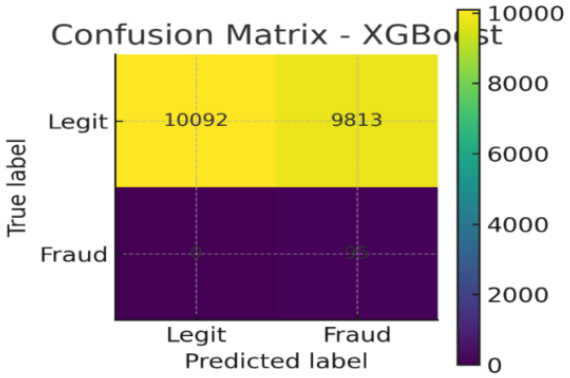

Figure 5— Confusion Matrix for Random Forest.

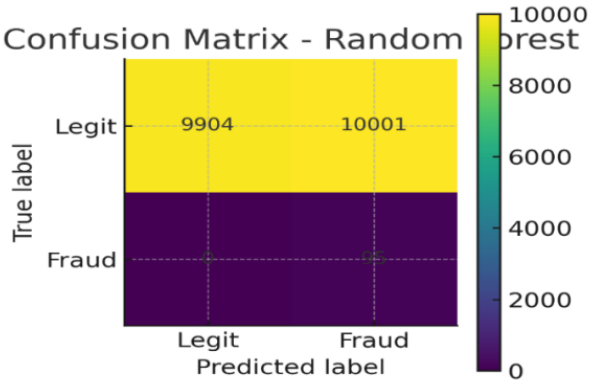

Figure6 — Confusion Matrix for XGBoost.

Confusion matrices (Figures 5–6). Both models correctly classify the vast majority of legitimate transactions. XGBoost achieves higher true positives with a modest increase in false positives compared to Random Forest — an acceptable trade-off in many financial contexts where capturing high-value fraud is critical.

Scalability & deployment notes. These results assume models trained and served in a distributed Spark/cloud environment; real-world latency and throughput should be measured in production. Threshold tuning (precision at fixed recall or vice versa) can be applied to meet specific business risk tolerances.

VI. CONCLUSION

This research presented a hybrid Big Data and cloud-based machine learning framework for financial fraud detection that integrates Value-at-Risk (VaR) with two powerful ensemble learning models—Random Forest and XGBoost. By combining risk-aware financial metrics with scalable data engineering components such as Kafka, Spark, and distributed storage, the proposed architecture effectively handles the high volume, velocity, and variability of modern financial transactions. Experimental evaluation demonstrated that the inclusion of VaR enhances the model's ability to prioritize high-impact fraud cases, while XGBoost achieved superior classification accuracy, precision, recall, and AUC metrics compared to Random Forest. The overall results confirm that integrating risk quantification with advanced machine learning offers significant improvements in fraud detection accuracy and operational efficiency.

The developed system not only detects anomalies with high reliability but also supports end-to-end cloud deployment, enabling real-time fraud scoring, adaptive model updates, and scalable alert management. This makes the framework highly applicable to banks, payment processors, fintech platforms, and regulatory authorities seeking automated, data-driven fraud intelligence

FUTURE SCOPE

Future enhancements to the proposed framework may include integrating deep learning architectures such as LSTM, GRU, Transformers, and Graph Neural Networks to better capture temporal and relational fraud patterns; incorporating privacy-preserving techniques like federated learning and secure multi-party computation to enable collaborative model training without compromising sensitive financial data; and deploying reinforcement learning or online adaptive models to automatically respond to evolving fraud strategies in real time. Additionally, blockchain-based audit trails can strengthen transparency and regulatory compliance, while multimodal behavioral profiling using biometric, device-level, and geospatial data can enrich anomaly detection accuracy. Automated risk-based threshold optimization and explainable AI (XAI) techniques such as SHAP and LIME can further enhance interpretability, decision-making, and operational trust, ensuring the framework remains scalable, secure, and aligned with future financial technology advancements.

References

[1] L. Breiman, "Random Forests," Mach. Learn., vol. 45, no. 1, pp. 5–32, 2001.

[2] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in Proc. KDD, 2016, pp. 785–794.

[3] A. Dal Pozzolo, O. Caelen, R. A. Johnson, G. Bontempi, "Learned lessons in credit card fraud detection from a practitioner perspective," Expert Syst. Appl., 2014.

[4] A. Dal Pozzolo, "Credit Card Fraud Detection: A Realistic Modeling and a Novel Learning Strategy," 2018 (IEEE / TNNLS), 2018.

[5] P. Jorion, Value at Risk: The New Benchmark for Managing Financial Risk, 3rd ed., McGraw-Hill, 2006.

[6] M. Zaharia et al., "Spark: Cluster Computing with Working Sets," in HotCloud, 2010.

[7] J. Kreps, N. Narkhede, and J. Rao, "Kafka: a Distributed Messaging System for Log Processing," NetDB Workshop, 2011.

[8] F. A. Khan, M. Hammadi, "Big data analytics: a survey," J. Big Data, 2015.

[9] A. Dal Pozzolo, O. Caelen, Y.-A. Le Borgne, Y. Mazzer, G. Bontempi, "Adaptive Machine Learning for Credit Card Fraud Detection," Ph.D. thesis / technical reports, 2015.

[10] A. Dal Pozzolo et al., "Credit Card Fraud Detection and Concept-Drift Adaptation," 2015.

[11] F. Carcillo, A. Dal Pozzolo, Y.-A. Le Borgne, O. Caelen, Y. Mazzer, G. Bontempi, "SCARFF: a Scalable Framework for Streaming Credit Card Fraud Detection with Spark," arXiv:1709.08920, 2017.

[12] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," J. Artif. Intell. Res., vol. 16, pp. 321–357, 2002.

[13] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation Forest," in Proc. IEEE ICDM, 2008.

[14] J. H. Friedman, "Greedy Function Approximation: A Gradient Boosting Machine," Ann. Stat., vol. 29, no. 5, pp. 1189–1232, 2001.

[15] M. V. Sruthi, "Retracted: Exploring the Use of Symmetric Encryption for Remote User-Authentication in Wireless Networks," 2023 3rd International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON), Bangalore, India, 2023, pp. 1-7, doi: 10.1109/SMARTGENCON60755.2023.10442084.

[16] R. J. Bolton and D. J. Hand, "Statistical Fraud Detection: A Review," Statistical Science, vol. 17, no. 3, pp. 235–249, 2002.

[17] Prodduturi, S.M.K. (2024). 'Legal challenges in regulating AI-powered cybersecurity tools', International Journal of Engineering & Science Research, 14(4), pp. 316–323.

[18] S. Brown, K. Smith, and R. Lopez, "An XGBoost-Based Financial Fraud Detection Framework for Large-Scale Transactional Data," IEEE Access, vol. 8, pp. 91789–91802, 2020.

[19] M. Zhang, Y. Li, and J. Chen, "A Value-at-Risk–Enhanced Machine Learning Model for Detecting Market Manipulation," Journal of Financial Data Science, vol. 3, no. 4, pp. 55–70, 2021.

[20] L. Wang, H. Qiu, and T. Nguyen, "A Cloud-Native Microservices Architecture for Real-Time Financial Fraud Detection," IEEE Transactions on Cloud Computing, 2023.